CMM OR CMMI, WHICH IS MORE APPROPRIATE FOR A SOFTWARE INDUSTRY?

*Monika Yadav

**Kaushik Kumar

ABSTRACT

It is not easy to improve software quality by relying on conformance to industry standards by continuously upgrading from one standard or model to another standard or model because this exercise is complicated for some software organizations. Many multinational companies, developed internal standards based on the military standards, and then sought to improve the standard even further as their software development processes matured. The software development systems based on these internal, commercial standards, and improved over the years have proved to be good systems. This paper shows how to adopt an efficient, workable system from basic principles that can improve the output of a software organisation by using CMM, CMMI.

Keywords: Analogy, Capability Maturity Model , Capability Maturity Model Integration, Exemplification.

[*]Research Scholar, Department of Management, Birla Institute of Technology, Mesra, Ranchi, India

[**]Associate Professor, Department of Mechanical Engineering, Birla Institute of Technology, Mesra, Ranchi, India

1.INTRODUCTION

A huge number of software standards, methodologies, practices, models and guidelines are introduced to current era of software engineering. These standards tend to be one size fits all approach that may be optimum for some projects but is often times ill-suited for others [1] because they are continually changing, which has become a complicated exercise for software industries, however, many companies developed internal standards based on the military standards, and then improved the process as their software development processes matured. The software development systems based on these internal, commercial standards, and improved over the years have proved to be good systems [2]. Carnegie Mellon developed Capability Maturity Model (CMM) as a process maturity model. Implementation of CMM raised many challenges that led to development of Capability Maturity Model Integration (CMMI) as an improvement. CMMI however does not replace CMM and the effectiveness depends on the specific area of application.

1.1 CAPABILITY MATURITY MODEL

The very first CMM was developed and released in August of 1990. The CMM for Software was developed by the Software Engineering Institute (SEI). The SEI CMM has been the standard set of quality guidelines for companies developing software for the U.S. Government i.e. usually the Department of Defense and it was initially developed as an assessment model for software engineering management capabilities of software providers .Companies wishing to develop software under these standards are evaluated according to five capability levels, ranging from uncontrolled development processes to consistently effective organization-wide implementation. This maturity model presents a growth theory according to which the quality level of a systems development organization can grow along a given growth path. The gist of the model is that several quality levels for the systems development process can be recognized. As a result of this deeper understanding, new practices in process-based software engineering have emerged in the last decade. This model came in picture due to aid the US government in evaluating software providers' abilities to handle large projects. Prior to the development of the model, many of the companies accomplished projects with considerable flaws in scheduling and budgeting. The model helped to solve these problems as CMM was originally developed for Software

Development and Maintenance. CMM Maturity Levels are discussed in many articles and case studies which can be depicted as shown in the table 1.

Table 1: Capability maturity model levels

S.NO	Level	Description
Level 0	**Incomplete Processes**:	Absence of processes or general failure to achieve process objectives. No process to product realization.
Level 1	**The Initial Level: Performed Processes**	The first and the lowest level in CMM is the *Initial* level. At this point organizations have few or no processes. Successes are mainly due to individual initiative and effort and processes that may exist are given a go-bye in crisis. The outcome of a project is therefore unpredictable.
Level 2	**The Repeatable Level: Managed Processes**.	At repeatable level, the processes are followed at the project level for various software project management functions and their performance is planned and tracked through a documented process. At this level, since the project management processes are in place, the organization is 'disciplined' and processes are expected to repeat successful practices as done in similar projects.
Level 3	**The Defined Level: Established Processes**	At this level, the organization defines processes for software engineering and management are standardized across the organization. Tailoring guidelines are developed to create project defined software processes and activities become stable and repeatable for implementing them organization-wide.
Level 4	**The Managed Level: Predictable Processes.**	It is reached when the organization uses quantitative goals for managing. Quantitative goals are set for software products and processes, using an organization-wide measurement program. The level involves a quantitative understanding of process capability and using this to manage processes. Variation in process performance is tracked and risks are identified and managed.
Level 5	**The Optimizing Level:**	It is the highest maturity level of the CMM. At this level, the organization improves continuously, setting new goals

	Optimized Processes	and responding to new technologies and challenges. Processes are cost-effective and are improved over time to meet the organization needs. At this highest level, the process performance is measured for continuous process improvement to verify whether the changes in the processes are providing the expected benefits.

1.2CAPABILITY MATURITY MODEL INTEGRATION

Currently, there are several maturity models, standards, methodologies, and guidelines that can help an organization improve the way it does business. However, most available improvement approaches focus on a specific part of the business and do not take a systemic approach to the problems that most organizations are facing. For example, maturity models such as the Software Engineering Institute's (SEI's) Capability Maturity Model for Software (SW-CMM), which focuses on improving software, and the Electronic Industries Alliance's (EIA's) Systems Engineering Capability Model (SECM), which focuses on systems engineering are available. By focusing on improving only one area of a business, these models have unfortunately perpetuated the barriers that exist in organizations [3].

Capability Maturity Model Integration (CMMI) provides an opportunity to avoid or eliminate these barriers through integrated models that transcend disciplines. CMMI consists of best practices in software companies. It addresses practices that cover the product's life cycle from conception through delivery and maintenance. There is an emphasis on both systems engineering and software Engineering and the integration necessary to build and maintain the total product [4]. It is an integrated model of many CMMs intended to achieve process improvement. CMMI has two representations Staged representation, Continuous representation [5-6].

Staged Representation CMMI pushes to increase the maturity of the processes, focuses improvement on the process capability an organization can expect to attain; however, this expected capability or ability to function in a mature manner is contained within maturity levels or stages. This representation provides a roadmap for sequencing the implementation of groups of process areas. There are five maturity levels, ranges from level 1 to 5 as shown in figure 1, with each level providing the foundation for further improvements.

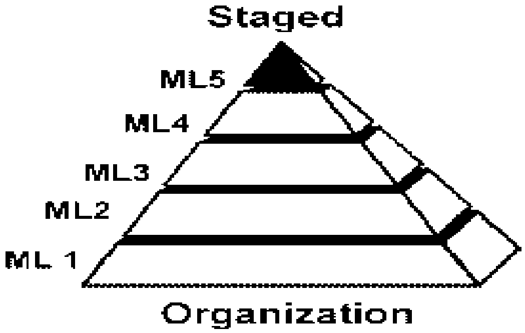

Fig 1: Staged representation of maturity levels in CMMI

Continuous Representation it has the same basic information as the staged representation, which arranged differently, provides maximum flexibility. In this each process capability level ranges from 0 to 5 which are depicted in figure 2.The continuous representation provides flexibility for selecting processes fit for achieving business goal of the organization [7].

CMMI provides 25 process areas means a cluster of related practices in these areas which are implemented collectively, satisfies a set of goals considered important for making significant improvement [8]. The CMMI model is a process that focuses on what to do, not how to do it or who does it. Gist of CMMI is to provide guidance for improving organization’s processes and ability to manage the development, acquisition, and maintenance of products or services.

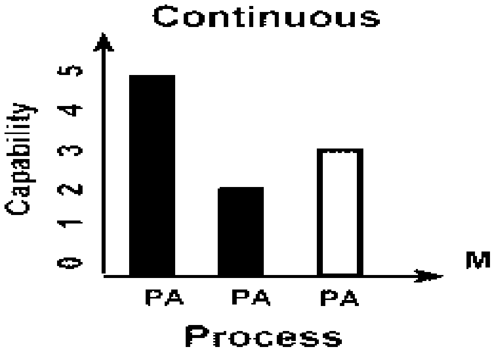

Fig 2: Continuous representation in CMMI

2. EXEMPLIFICATION OF CMM AND CMMI

CMMI was developed to build on the best practices of CMM. It has been felt by some that CMMI will be successful where CMM could not because most organization that implemented CMM used while still entrenched in a default waterfall mentality. Both Software CMM and CMMI models are based on the pretext that organization will follow process improvement journey in small incremental steps rather than bringing radical change through large scale sweeping changes. Quality improvement through reengineering can be bought through department wise small evolutionary steps and by repeating small wins successively across the organization As asserted by Paulk and others [4, 9-10] that software CMM and CMMI (staged) quality improvement models provide a baseline for incremental SPI by defining five maturity levels that lay down a framework with measurement and assessment criteria for an organization's software process maturity and for assessing its SPI capability. CMM and CMMI both are supported by case studies and data that they promote Return on Investment [11]. According to some researchers CMM is a logical approach and common sense for Software engineering and quality improvement practices [3, 6]. Software engineering should be done to achieve business and organizational goals one should not get into the debate that which model is better.

Table 2: *Mapping of capability maturity model CMM /capability maturity model integration CMMI*

S.No	CMM	CMMI
1	CMM model is superseded by CMMI CMM is a reference model of matured practices in a specified discipline like Systems Engineering CMM, Software CMM, People CMM, Software Acquisition CMM etc. But they were difficult to integrate as and when needed.	CMMI is the successor of the CMM and evolved as a more matured set of guidelines and was built combining the best components of individual disciplines of CMM (Software CMM, People CMM etc). It can be applied to product manufacturing, People management, Software development etc.
2	CMM describes about the software engineering alone where as CMM	CMMI also incorporates the Integrated Process and Product Development and

	Integrated describes both software and system engineering.	the supplier sourcing.
3	The CMM model proved useful to many organizations, but its application in software development has sometimes been problematic.	The Capability Maturity Model Integration (CMMI) project was formed to sort out the problem of using multiple CMMs.
4	CMM measures the maturity level of an organization by determining if an organization completes the specific activities listed in the Key Performance Areas (KPA), oblivious to whether the completion of such activity leads to the desired result.	CMMI is also an activity based approach but the major difference is that CMMI takes a more result-oriented approach when defining and measuring Key Performance Areas.
5	CMM KPA concentrates on the completion of specific tasks or processes and does not motivate the organization to focus on process architecture.	CMMI, on the other hand has an iterative lifecycle that integrates the latest best practices from the industry and attacks risks in process architecture at an early stage.
6	CMM is still relevant and appropriate for sequential, activity-based management paradigm.	CMMI supersedes CMM in software development processes.
7	CMM is concerned at recording processes.	CMMI documentation and meetings focus on strategic goals of the organizations.
8	Initially, CMM describes specifically about software engineering.	CMMI describes integrated processes and disciplines as it applies both to software and systems engineering.
9	CMM has a defined path for over all organizational improvement	CMMI continuous grants freedom in improving only those process areas which are critical for organization to improve and mitigate

		the risk
10	CMM focuses on set of processes to achieve SPI by defined Maturity Levels	CMMI Continuous enables increased foresight into process area capability improvement
11	CMM controls the pace of improvement based on maturity level	CMMI allows improvement in different processes at different rates and gives cost and time flexibility to organization

3. ANALOGY OF CAPABILITY MATURITY MODEL I AND CAPABILITY MATURITY MODEL INTEGRATION

As we know that CMM and CMMI is designed specifically for software industry there is some features are common in between these quality standards which are depicted in the table no 3.The common feature in these two is that they gave importance to paperwork and meetings that distract management's time and effort from actual work process.

Table 3: *Analogy of CMM capability maturity model/capability maturity model integration CMMI.*

Level	*CMM And CMMI*
Level 1 (Initial):	The first level of both CMM and CMMI describes an immature organization without any defined processes, run in an ad hoc, uncontrolled, and reactive manner.
Level 2 (Repeat):	Organizations that repeat some processes attain Level 2 CMM. Level 2 of CMMI however requires management of organizational requirements through planned, performed, measured, and controlled processes in standards, procedures, tools, and methods.
Level 3 (Defined):	CMM Level 3 mandates a set of documented standard processes to establish consistency across the organization. CMMI Level 3 is an improvement of CMMI Level 3 and describes the organizational process.
Level 4 (Managed):	CMM Level 4 requires organizations to attain control over processes by using quantitative statistical techniques. CMMI Level 4 demands likewise, but also identifies sub processes that significantly contribute to overall process efficiency
Level 5 (Optimized):	CMM Level 5 mandates use of quantitative tools and objectives to manage process improvement. CMMI Level 5 on the other hand focuses on continuously improving process performance through incremental and

	innovative technological improvements

.

4. Conclusion

So it can be observed that although the levels of both CMM and CMMI are similar but the difference lies in perception of these. CMM is a standard where as CMMI integrates different CMM models and creates an integrated CMM customised for the specific organisation. Hence the CMMI has provided significant value for many organizations that have used it as a guide for improving the way they do their engineering work. It has helped them to gain control over their processes—Management, Engineering, and supporting processes to assure that those processes serve the needs of the organization. Since CMMI is a customised standard hence by following the same more consistent success in the engineering projects can be achieved by putting the organization on the road to more effective processes. Most software organizations can start with CMM then add CMMI. When a software organization aims to achieve software process improvement, it is necessary to address more aspects, likes CMMI.

REFERENCES

1. Software Quality Professional: VOL. 8, Issue Two/© March 2006, ASQ publication: Page.28, “Are Standards the Answer?”

2. Handbook of Software Quality Assurance edited by G. Gordon Schulmeyer and James I. McManus. – 3rd ed-1998. : Page.104, “Standardization of Software Quality Assurance – Conclusion”

3. Royee ,walker vice president and general manager of strategic services of rational software corporation ,the rational edge,CMMvs CMMI from conventional to modern software management feb 2002.

4. Mary Beth Chrissis, Mike Konrad, Sandy Shrum. ”CMMI: Guidelines for Process Integration and Product Improvement”. 2003. Addison Wesley Professional

5. Software Engineering Institute (SEI). ”CMMI version 1.1 CMU/SEI-2002-TR- 012”. 2002.

6. Richard Basque. ”CMMI un itin´eraire fl´ech´e vers le Capability Maturity Model Integration”. 2004. Dunod.

7. B. Mutafelija and H. Stromberg, Systematic Process Improvement Using ISO 9001:2000 and CMMI, Artech H.

8. M. B. Chrissis, M. Konrad and S. Shrum, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003.

9. Mark C. Paulk & ldquo. A Comparison of ISO 9001 and the Capability Maturity Model for Software,&rdquo, Tech. Report CMU/SEI-94-TR-12, Software Eng. Inst., Pittsburgh, July 1994.

10. Herbsleb, J., Carleton, A., Rozum, J. Siegel, j, and Zubrow, D. Benefits of CMM-based software process improvement: Initial results. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA CMU/SEI-94-TR- 13, Aug. 1994.

11. Carnegie mellon software engineering instittute.http://www.sei.cmu.edu. “upgrading from SW-CMM to CMMI”